IMS Newsletter 4
(1981) 24-27

Roots of Real Polynomials

W.J. Kirk

and

A.G. O'Farrell

The simplest sure-fire way to approximate the real zeros of a real polynomial is to use Sturm's theorem. Horner's method misses double zeros, and Newton's method does not always work. The method based on Vincent's theorem [1] is very elaborate.

Sturm's theorem is not usually covered in basic algebra courses, even though it is in van der Waerden [2, p. 284] . Consequently, it is not as well-known as it should be. It provides a straight-forward, computationally effective method for approximating the real zeros (without multiplicities) of a real polynomial. It goes as follows.

Let $f(x)$ be a polynomial with real coefficients. Let $F_0 = f$, and $F_1 = f'$ (the derivative of f). Carry out the following variant of the Euclidean algorithm:

$$F_0 = F_1 Q_1 - F_2$$

$$F_1 = F_2 Q_2 - F_3$$

$$\cdots\cdot$$

$$F_{m-2} = F_{m-1} Q_{m-1} - F_m$$

$$F_{m-1} = F_m Q_m .$$

Note that the customary plus sign has been replaced by a minus sign. Equivalently, carry out the usual Euclidean algorithm, and change the sign of the j-th remainder whenever j is congruent to 2 or 3 modulo 4. For any real number x, let $n(x)$ denote the number of sign changes in the sequence

$$F_0(x),\ F_1(x),\ \ldots,\ F_m(x)$$

(where terms equal to zero are omitted, thus $+,0,+$ is not a sign change, and $+,0,-$ is). This sequence is called the Sturm chain. Sturm's theorem states that if $a < b$, then the number of zeros of f in the closed interval $[a,b]$ is $n(a) - n(b)$.

This algorithm is easily programmed. Polynomials can be treated as vectors (with the coefficients as components), and one can write simple routines to carry out polynomial arithmetic (additon, multiplication, division, and Euclidean algorithm). Once the Euclidean algorithm is programmed, it is easy to write a routine to calculate $n(x)$, for any given x (since there is an algorithm for differentiating polynomials). A routine to find all the roots in the interval $[a,b]$ proceeds by checking $n(a) - n(b)$, then bisecting the interval, checking n at the endpoints, and continuing. At the n-th step, the positions of the roots are known to within $|a-b|/2^n$.

To find all the roots on the line, use the (obvious) fact that all the roots of

$$f(x) = a_n x^n + \ldots + a_0$$

lie in the interval $[-s,s]$, where

$$s = 1 + \max_{0 < j < n-1} \left| \frac{n a_j}{a_n} \right|^{1/(n-j)} .$$

To detect multiple roots, examine the zeros of f' .

We wrote a BASIC program to perform the Sturm algorithm. Apart from solving equations, it is useful for demonstrating the Euclidean algorithm for polynomials. It uses about 8k words of core on the DEC PDP 11/34. If necessary, it can be pruned considerably to fit in smaller workspaces. Copies of the listing may be had by writing to the authors.

On our time-sharing system, the program finds the total number of roots in a few seconds. The time taken to locate r roots with error less than h is roughly proportional to $r \log_2 h$. The constant of proportionality is less than 6 seconds.